

EXHIBIT 21

MAGIC Final Report

Yvan G. Leclerc, SRI International

This is an HTML version of the final report submitted to DARPA in May, 1996 for SRI's part in the MAGIC project.

1. Introduction

The Multidimensional Applications and Gigabit Internet Consortium (MAGIC) project was a collaborative effort involving many participants, including the Earth Resources Observation System Data Center (EROS Data Center, or EDC), the Lawrence Berkeley National Laboratory (LBNL), the Minnesota Supercomputing Center Inc. (MSCI), Mitre Corp., Sprint USA, and the University of Kansas (KU).

This report summarizes SRI's contributions to this collaborative effort. We make extensive use of World Wide Web (web) URLs to point to more detailed information for the interested reader. For an overview of the MAGIC project, see the [MAGIC Home Page](#), and in particular the [MAGIC Overview document](#).

Much of the technical information in this document can be found through SRI's [MAGIC Home Page](#).

The primary objective of SRI's contribution was the design and implementation of an application that allows a user to interact in real time with a synthetic 3D photo-realistic view of a large terrain. This application, called TerraVision, combines elevation data, aerial photographs, models of buildings, and models of vehicles whose positions were obtained using GPS receivers, all stored in a remote terrain database accessed via a highspeed network.

The terrain database is stored remotely on a parallel distributed storage system called the [Image Server System \(ISS\)](#) implemented by LBNL. As the user moves over the terrain, the relevant parts of the terrain database are accessed across the network in real time. Thus, a user can roam over arbitrarily large databases without having to wait for the entire database to be downloaded first. Because of the finite delay in accessing data over networks (even very high-speed networks such as the MAGIC network), a local cache of the visible and near-visible portions of the terrain database is maintained, so that the user can move quickly with no apparent delay. Other techniques for maintaining high image quality and small user delays in the face of uncertain network speed and latency are discussed in Section 3.

The application's architecture has been very successful. TerraVision has been demonstrated on many occasions at Sprint's headquarters and elsewhere, and has been installed several sites, including DARPA's Enterprise Room (see Section 4).

The secondary objective of SRI's contribution was the design and implementation of a system for creating large terrain databases for use by TerraVision. A system based on *LISP for a massively parallel Connection Machine, called TerraForm/CM, was implemented that allowed us to create one of the largest seamless databases in the world, covering approximately 40 X 25 km of Ft. Irwin, CA at 1 meter resolution. Because of the demise of Thinking Machines Corp., we were forced to abandon the *LISP-based system in favor of a distributed processing approach. Within the limited resources available to us at that point in the project, we created a "proof-of-concept" system based on SRI's GLU distributed processing environment to demonstrate the feasibility of creating a full-scale data-preparation system using this approach.

In the following sections, we describe the terrain database, data preparation processes, and TerraVision, followed by a description of demonstrations and technology transfer resulting from our project.

12/23/2004

2. Data Preparation (TerraForm)

TerraVision uses aerial (or satellite) images combined with elevation data to create realtime synthetic 3D views of a site. The imagery and elevation data must be preprocessed in several ways before they are ready for storage on the ISS and used by TerraVision. In particular, perspective effects must be removed in a process called ortho-rectification, the multiple overlapping images covering a site must be combined into a single seamless image called a mosaic, and a multiresolution pyramid of the mosaic must be created.

TerraVision also needs certain information relating the final processed imagery and elevation data to real-world coordinates (latitude and longitude) so that building models and vehicles can be correctly positioned. TerraForm (the rubric for the suite of programs used in processing the imagery) requires additional information about the imagery, such as the exact position, orientation, and focal length of the camera, for the orthorectification and mosaicking processes.

The collection of processed imagery, elevation data, and coordinate information for a given site is called a GeoPyramid tile set. Each individual image is called a Pyramid tile set. The information files describing a tile set are called tile set specification, or tspec, files. For more information, see [Tile Sets](#) and the [MAGIC Project](#).

A Pyramid is a multiresolution representation of a single image, be it an original (digitized) aerial image, a processed satellite image, or an image representing elevation data (also called a Digital Elevation Model, or DEM). Each Pyramid forms a resolution hierarchy, or pyramid, which is a series of images of increasingly lower spatial resolution. For example, most images in this project correspond to approximately 1 meter ground resolution (that is, each pixel in the image covers approximate 1 meter square on the ground). This first image, or Pyramid level, forms the base of the Pyramid. The next level in the series corresponds to 2 meter ground resolution, and requires half as many pixels in each direction. The next level is at 4 meters, and so on.

Each level of a Pyramid is divided into small, equal-sized subimages called tiles. These tiles are stored as independently accessible items on the ISS, or as independent files in a local file system. TerraVision requests these tiles from the ISS by specifying the level and the (x,y) coordinate of the tiles that it needs.

A GeoPyramid is a collection of Pyramids corresponding to a single site. It contains a DEM, a mosaic of ortho-rectified images, and optionally, the original images used in the creation of the mosaic.

During this project, we have created several dozen Pyramids and GeoPyramids in collaboration with the EDC. These tile sets range in size from 9K X 9K pixels for a single aerial or SPOT image, to 39K X 25K (approximately 1.25 Gbytes) for a mosaic of 30 aerial images. A simple database of these tile sets was created and is accessible over the web using the [tsmWeb](#) interface, where a user can browse through the database and view parts of any Pyramid.

The tiles, database, and tspec files are accessible in two ways: either through the tsmWeb web site mentioned above, or by using a C library, written by SRI, called the [Tile Set Manager API \(tsmApi\) library](#). The tsmApi library allows applications, such as TerraVision, to access tile sets stored either on ISSs, simple file systems, or on an HTTP server.

The tsmApi library was written to provide a simple uniform mechanism for applications to read and write tile sets, independent of the storage medium. This library has proven to be invaluable because it allowed us to continue development of both TerraVision and TerraForm during the many times that network connections to a local ISS were not functional, or that the ISS was not working correctly. In retrospect, we probably should have written this library at the very beginning of the project, because we spent a great deal of effort dealing with the instability of network connections and the ISS throughout most of the project.

12/23/2004

The tsmApi library is also used by LBNL to load a tile set from a local file system onto the ISS.

2.1 TerraForm/CM and TerraForm/CME

When this project first started, it was thought that a Connection Machine (CM) located at MSCI would be used to provide near-real-time processing of incoming imagery. A CM is a massively parallel supercomputer that showed great promise for this task. Unfortunately, a number of insurmountable problems made this goal impossible. However, we did use one of the CMs (a CM-200) to process 30 NAPP aerial images of Ft. Irwin (supplied by EDC), each approximately 9K X 9K pixels in size, into an orthorectified mosaic of approximately 39K X 25K pixels, or 1.25 Gbytes. At the time it was created, this was probably the largest seamless dataset of its kind in the world.

The data preparation system running on the CM was written in *LISP, a data-parallel extension of the LISP programming language. The initial processing system was written in LISP to take advantage of SRI's Cartographic Modeling Environment (CME), a system used to create complex site models from aerial and satellite imagery. To test this system, called TerraForm/CME, an orthorectified mosaic of 36 low-resolution (16 meter resolution) images of Ft. Irwin was successfully created in late 1993. This test demonstrated that the architecture and processing steps were correct. In addition, CME was used to create and place approximately 500 building models on the site. It was also used to verify the placement of the vehicles whose positions were determined using GPS receivers and stored using GATS, a system developed by SRI (see Section 3.4).

Since *LISP is an extension of LISP, we were able to create a LISP version of the system by replacing some of the low-level I/O and processing functions in TerraForm/CME with *LISP functions (the great majority of the LISP code is common to both systems). This system, called TerraForm/CM, was completed in early 1994 using SRI's CM as the development platform. TerraForm/CM was then installed on the CM at MSCI and, in collaboration with EDC, it was used to ortho-rectify and mosaic 30 aerial images (stored at EDC) to form the 1.25 Gbyte dataset mentioned above, completed in mid-1994.

Unfortunately, overall processing times were very high because the ISSs were not available for data storage at EDC, and I/O times dominated (i.e., actual processing time on the CM was only a small fraction of the overall time required to read, process, and write each image).

Although the overall processing times on the CM did not bode well, it was thought that with the introduction of ISSs at EDC and the faster CM-5, processing times would become acceptably small. Unfortunately, Thinking Machines Corporation went into Chapter 11 at about this time. Consequently, its developer stopped working on *LISP for the CM-5. MSCI also returned its CM-200, which left the project without a platform for using TerraForm/CM.

The disappearance of a *LISP-capable platform forced SRI and the members of MAGIC to investigate other avenues for high-speed processing.

2.2 TerraForm/GLU

SRI and the other MAGIC participants decided that a distributed processing approach to data preparation would not only yield a system that could process imagery at the requisite rates, but would also prove to be an excellent test of the ISSs and the MAGIC network.

Although at this point (early 1995) we did not have the resources to create a completely new version of TerraForm, it was decided that creating a proof-of-concept application that exercised all the steps in the data preparation process would be useful.

After some research, we decided to base the proof-of-concept application on SRI's Granular Lucid (GLU)

12/23/2004

distributed processing environment.

In the GLU environment, a processing task is divided into two parts: a generator process (running on one processor) and many worker processes (typically one worker process per processor, although more than one might be appropriate for fast processors). The code for the generator process is written in a mixture of Lucid and C, while the worker code is written entirely in C. In fact, the worker code is typically a C function written with no particular knowledge that it will run in the GLU environment, making it possible to use standard C libraries.

Within the GLU environment, all the data preparation tasks have a common architecture:

- There is a specified input tile set on a specified ISS.
- The generator process allocates space for an output tile set on a specified ISS.
- The generator process divides the set of input tiles into a number of square groupings of tiles (say 8 X 8 tiles).
- Once divided in this manner, the GLU system sends the specifications of the tile groupings to the worker processes as they become available. Worker processes that crash or that become disconnected because of networking problems are automatically handled by GLU.
- Each worker process takes as input the specification of a tile group, reads those tiles from the ISS (or the specified storage medium), processes them, and then writes the processed tiles into the newly created output tile set on the ISS.
- When a worker process is finished with a group of tiles, it waits for a new tile group from the generator.

In this architecture, the GLU generator program never communicates directly with the ISS and does very little processing of its own. Consequently, this single process is unlikely to become a bottleneck in the processing. All processing and communication with the ISS happen in the worker processes. Since there can be as many of those as there are available processors on the MAGIC network, we can expect that a completed TerraForm/GLU would make full use of the network and ISS.

To start a GLU system, one needs to start the generator process on some "master" host, and the worker processes on as many worker hosts as desired (when a worker process is started, it is provided with the IP address of the master host). The GLU environment provided no mechanism for starting worker processes other than by manually logging into the hosts and starting the worker. This manual procedure was inappropriate for the MAGIC project, since the worker hosts were in different administrative domains, and providing a username and password to anyone wishing to process imagery was not feasible.

To overcome the startup problem, we developed a general mechanism for starting a GLU system based on HTTP servers. This new startup procedure begins by having a user select a dataset for processing using the tsmWeb web site mentioned earlier. Then the user clicks on a button to access the "GLU Distributed Processing Manager" form. At this point, the user selects the type of processing to be done, the hostname for the generator process, and the hostnames for the generator processes. Once this is done, the user clicks on the Submit button to start the master and generator processes.

Clicking on the Submit button of the "GLU Distributed Processing Manager" form starts the Distributed Processing Manager on the HTTP host. The Manager (a CGI program) must then start the generator and worker processes on the GLU hosts selected by the user. For those GLU hosts that are in the same administrative domain as the HTTP host, the Manager can start the GLU processes directly, using the system "rsh" command. For other GLU hosts, the Manager must use an intermediate host, called the "dispatch host" associated with the GLU host. The Manager sends an HTTP request to each dispatch host to start a Distributed Processing Manager. It then provides each of these Managers the list of GLU processes that should be started at that site. (See the [GLU Manager Home Page](#) for more details of the dispatching process.)

12/23/2004

Once the GLU processes are started, the user's browser is redirected to a web page showing the progress of the GLU processing.

Using the above architecture and startup mechanism, we implemented a "check-sum" process that reads in tiles and sums together all the bytes in all the tiles (modulo 230). This process was ported to EDC and to Sprint's TIOC headquarters, and a number of tests were run.

This "proof-of-concept" process is quite useful for a number of reasons:

- It exercises all aspects of GLU that would be required for full processing.
- It exercises the GLU startup mechanism.
- It exercises the communication between GLU workers and the ISS.
- It verifies the correctness of all of the above by comparing the resulting sum against the known correct sum. If any part of the computation or communication fails, or if the data was incorrectly stored on the ISS, the sum will be incorrect.

3. Terrain Visualization Application (TerraVision)

TerraVision is a high-speed graphics application that allows a user to interact in real time with a synthetic 3D photo-realistic view of a large terrain. It combines elevation data, aerial photographs, models of buildings, and models of vehicles whose positions were obtained using GPS receivers, all stored in a remote terrain database (ISS) accessed via a high-speed network.

TerraVision was designed to run on certain Silicon Graphics (SGI) workstations to take advantage of their high-speed graphics hardware. When this project started, the only moderately priced platform that could support both the high network bandwidth and requisite graphics performance was a multiprocessor Onyx workstation. SRI exercised one of its options to purchase a four-processor SGI Onyx workstation as a development platform (each processor is a 150-MHz R4400). Sprint purchased a more powerful sixprocessor Onyx to demonstrate TerraVision (and other applications) at its Technology Integration and Operations Center (TIOC).

3.1 TerraVision Design Goals

Any interactive graphics applications such as TerraVision must meet several critical design constraints:

- High refresh rates: images should be refreshed at a rate close to 30 frames per second, with 10 frames per second being the lowest usable frame rate.
- Low latency: there must be very little delay between a user's mouse/track-ball movement and the corresponding effect on the image.
- High image quality: images should look as realistic as possible within the above constraints.

TerraVision's design goal was to provide a good compromise between these constraints in the face of uncertain networking/ISS speeds and latencies. To understand how the design goals were met, we first must understand the basic steps required to display a synthetic scene.

3.2 Overview of TerraVision Process

To display a synthetic scene, a graphics application needs two things: a mathematical model of the objects in the scene (their size, shape, color, and texture, typically represented as a collection of triangles and associated images called "texture maps") and a mathematical model of a synthetic camera (its position, orientation, field of view, and focal length). Interactive applications typically allow a user to change the position and orientation of the camera by using a mouse or trackball. Ideally, the application should refresh the screen 30 times per second, and mouse movements should take effect within one frame.

12/23/2004

To make perfect synthetic images, the entire model of the scene must reside in the computer's main memory. But it is just as important that the complexity of the scene be limited; otherwise, the graphics hardware will be unable to display the scene within the requisite time. For example, in the case of an SGI Onyx, a scene must be composed of no more than about 10,000 triangles, and the texture maps can occupy no more than about 4 Mbytes. Our current largest terrain database, however, comprises about 1 million triangles and 1 Gbyte of texture maps.

TerraVision was designed specifically to meet the design goals stated earlier given the large size of the terrain databases and the inherently uncertain delay in remotely accessing the databases. The design goals were met as follows.

3.2.1 Design Goal 1: Maximize Frame Rates

A number of interacting design decisions were made to meet rates of 10 to 30 frames per second.

The first design decision was to store the terrain database as a collection of individually accessible, equal-sized "tiles". Because they are individually accessible, TerraVision can retrieve and display only those tiles that are visible for a given viewpoint, thereby reducing the scene complexity.

The second design decision was to represent the terrain using a multiresolution hierarchy. This is necessary because, for example, a high-altitude view makes the entire scene visible and would otherwise require all the tiles. A multiresolution representation allows TerraVision to use different resolutions of tiles to display different parts of the terrain depending on the viewpoint.

In a multiresolution hierarchy, each tile occupies the same number of bytes in storage. But the different resolutions in the hierarchy represent different spatial extents. For example, image tiles are typically 128 X 128 pixels. At 1 meter resolution, an image tile covers 128 meters X 128 meters. At the next level of resolution (2 meters) an image tile covers 256 meters X 256 meters, or 4 times the area, and so on. This hierarchy continues until a single tile covers the entire scene.

By using high-resolution tiles in the foreground and low-resolution tiles in the background, a very large area can be displayed with a relatively small number of tiles, keeping the scene complexity at a minimum.

The third design decision was to separate the display procedure into two separate threads (a thread is an independent and separate "lightweight" process that can run on a separate CPU). The first "tile visibility" thread determines what tiles are visible and stores this set of tiles (after some preprocessing) in a triple-buffered display list in shared memory. Asynchronously, the second "rendering" thread picks the next available display list and displays the tiles from that list. This architecture allows two CPUs to be working in parallel. Furthermore, the rendering thread does little more than transmit triangles and texture maps to the graphics hardware, which makes optimal use of the SGI graphics hardware.

The tile visibility thread is, in some sense, the heart of the TerraVision system. It uses a coarse-to-fine search strategy that allows it to find all the visible tiles at the appropriate resolution in very little time. In particular, the time required to find the set of visible tiles grows only logarithmically with the number of tiles in the database. Thus, we can expect TerraVision to scale to databases that are 10 to 100 times as large as our current largest database with no need for faster hardware.

In brief, the coarse-to-find search strategy works like this (see TerraVision: A Terrain Visualization System for technical details). First, the lowest-resolution tile (covering the entire site) is examined to see if it is visible from the given viewpoint. If so, the next step is to see if it is at the appropriate resolution. If so, the tile is used at that resolution and the procedure is complete. If not, the tile is marked as "visible", and the four higher-resolution tiles (covering the same spatial extent as the single lower-resolution tile) are first checked to see if they are currently in memory. If not, the procedure is complete. If so, each tile is examined as the first one was. This procedure is applied recursively until no higher-resolution tiles need to

12/23/2004

be examined.

3.2.2 Design Goal 2: Minimize Latency

Since the terrain database is much too large to be kept in main memory, TerraVision keeps a small fraction of the available tiles in a local cache. By using a local cache, TerraVision is able to display a new view at any time, no matter how quickly the user moves. The penalty that is paid is that if the user moves too quickly (or jumps to an entirely new location), the display may not be at the desired resolution (see Section 3.2.3.)

The local cache is effectively a least recently used queue, maintained and replenished as follows.

First, the visible tile thread discussed in Section 3.2.1 marks all tiles that are required for a view as being "visible" and moves them to the head of the queue.

Second, the processes that retrieve tiles (discussed later) add newly received tiles at the front of the queue. If the queue was full, the tile at the end of the cache is removed.

Since the local cache is only a small fraction of the entire database, TerraVision must attempt to have all tiles required for the current viewpoint in memory for use by the visible tile and rendering threads. If the expected delay between sending a tile request and receiving the tile is D seconds, then TerraVision should predict the user's movements D seconds in advance, request the tiles from that predicted viewpoint (minus those already in memory), and store them in a local cache when they arrive.

Prefetching tiles, as this procedure is called, is carried out in two separate threads. There is a "tile prediction" thread for predicting the user's movement and determining the set of tiles that will be visible in the predicted viewpoint, and a "tile requesting" thread for requesting those tiles from the ISS (eliminating those that are already in memory). Both of these threads run at fixed intervals.

In addition, TerraVision uses a separate "tile receiver" thread for receiving tiles from each ISS server. These threads are free-running. They get the position of the next available empty slot in the tile cache and do a "blocking read" of the next tile from the server. When the tile arrives, the thread starts over again.

As with the visible tile thread, the tile prediction thread uses a coarse-to-fine search strategy. Consequently, the coarsest tiles are placed in the request list first. This placement is important because it means that, on average, the coarsest tiles (covering the largest spatial extent) will be received first.

An interesting consequence of the coarse-to-fine request strategy is that TerraVision can also run over relatively slow networks. As the user moves around the terrain over a slow network, TerraVision continues to display at its normal rate. But because the network is slow, by the time a coarse tile has arrived, the user has moved to a new location, so a different coarse tile is requested. Consequently, as the user moves the terrain, he or she sees only a coarse-resolution view of the terrain. If the user stops, eventually all of the tiles are delivered and the user sees a high-resolution view of the scene. On a high-speed network, such as the MAGIC network, the user usually sees the high-resolution view.

3.2.3 Design Goal 3: Achieve High Image Quality

This last design goal is to maintain the highest image quality possible given the tiles available in the local cache. Several strategies could be adopted for this.

One strategy would be to wait until all tiles required for a view are available before rendering. This is probably the simplest strategy, but it would conflict quite strongly with the minimal latency criteria because the delay between requesting a tile and receiving it can range from 0.1 to 0.5 second (or more on a slow

12/23/2004

network).

The strategy we have adopted is to use the coarse-to-fine strategy outlined earlier. The tile visibility thread attempts to find the appropriate resolution tile by iteratively dividing tiles into four (it actually divides the space represented by the tile into four). When a tile must be subdivided it checks to see if the four higher-resolution tiles are in memory. If not, then that tile is used, even though it's at a lower resolution than required. Consequently, the user sees the highest-resolution representation possible at all times.

3.3 Building Models

In addition to the terrain and aerial images, TerraVision can display a few simple building models. We used SRI's Cartographic Modeling Environment (CME) and the elevation data and aerial imagery provided by the EROS Data Center to position about 500 building models in the database. From a combination of building models provided by TEC and building models SRI developed from ground-level photography at Ft. Irwin, these sites are rendered as full 3D buildings on top of the terrain.

In the flyovers of the terrain, the buildings are rendered as 3D models. With ground-level photography taken at Ft. Irwin, some of the models have photo-realistic textures. In the overhead view of the terrain, the models are rendered as flat surfaces. The building models can be turned on and off at the user's discretion.

The rendering of the models decreases the frame rate of TerraVision because of the increase in the number of polygons and textures required to render the models. This drop in frame rate is especially noticeable when the user is flying over Ft. Irwin, which has the highest concentration of building sites.

3.4 GATS

When the MAGIC project was first proposed, it was thought that we would have access to training exercises at Ft. Irwin (the Army's National Training Center) so that we could visualize a live exercise using TerraVision and live feeds of vehicle positions using SRI's GATS (GPS/ATCCS (Army Tactical Command and Control System) Tracking System). Unfortunately, we were not able to obtain access to live exercises. Instead, SRI developed a system to record the GPS positions of vehicles and play these back to TerraVision in real time. SRI used this system to record tracks of a HMMWV (High Mobility Multipurposed Vehicle) at Ft. Irwin.

The playback system, called the "GATS server" here, typically resides on the MAGIC network. The user connects to the GATS server and selects from a list of prerecorded sessions. Once a file is selected, the vehicle positions are played back at the same rate as they were recorded.

SRI developed a vehicle model in the form of a HMMWV to represent the vehicle information. Errors inherent in the GPS data cause the vehicle models to not exactly line up with the underlying aerial imagery. Around the vehicle model, a yellow circle, which represents the typical error rate of the GPS used to record the data, is drawn to allow the user to visualize the approximate location of the real vehicle.

As many as ten vehicles at a time have been played back using this system. If real data are ever acquired and are in the same format, TerraVision should be able to render them in real time on top of the terrain.

3.5 TeReVision

TeReVision was developed by SRI to allow remote users to view and control TerraVision. Because TerraVision was developed for an SGI Reality Engine, it does not perform at interactive rates on the lower end SGIs nor on any other platform, such as a Sun. TeReVision attempted to address this issue by allowing the user to run a client on either a Sun or SGI and connect to a running TerraVision session.

12/23/2004

A user starts TeReVision by connecting to a preexisting TerraVision session. The communication is handled via a well-known port. Once communication is established between TeReVision and TerraVision, the remote user can now see and control TerraVision. The user interface for TeReVision is the same as on TerraVision, to give the remote user the ability to control all aspects of TerraVision. When the remote user performs an action, the command is transmitted to TerraVision, which acts on the command. The resulting imagery is sent back to TeReVision, which is then converted to an X11 image for display.

Because all the rendering in this system is done on the Reality Engine and remote from the user, problems arose because of latency issues. This was especially noticeable on congested or lossy networks. The remote user could click on a button and not see the results of the action until TerraVision had returned a new frame. This led to problems where the remote user got out of sync with the actions. One solution was to delay all user interaction until TerraVision had updated the remote user with a new frame. This caused lockouts where the remote user clicked on an action and the action was ignored.

Despite some of these problems, which are inherent in any visualization application where the rendering is done remotely, TeReVision was successfully demonstrated on the MAGIC network with the remote user residing on another part of the network and thus controlling TerraVision remotely.

3.6 XCPS

One of the difficulties that we faced with the development of TeReVision is that any modifications to TerraVision required corresponding modifications to TeReVision so that users would see the same interface when using either system.

To alleviate this problem, we used SRI's XCPS system to develop a proof-of-concept shared TerraVision system. XCPS is an X11 "Pseudo Server", which simply means that an application connects to XCPS instead of the normal X11 window server. XCPS then replicates the X11 commands and sends these to one or more XCPS servers on other machines, which in turn send them to their screens. In other words, users on other machines see exactly what the users of the original application see. Furthermore, mouse movements and clicks are transmitted back to the application just as if the original user were moving the mouse.

XCPS can be used as above to enable several viewers to share a view of a single application. An application using only X11 graphics commands does not need to be modified in any way for this to work. TerraVision uses a mixture of GL and X11 graphics commands, so it needed to be modified slightly. It was demonstrated at the MAGIC Symposium in August 1996.

3.7 The CAVE Environment

At DARPA's request, SRI created a special turnkey version of TerraVision that used the CAVE graphics environment so that TerraVision could be demonstrated at DARPA's Enterprise Room. This version of TerraVision was delivered to DARPA in late 1995.

4. Demonstrations and Integration Tests

TerraVision was demonstrated at numerous locations at different conferences and symposia over the course of the project. These demonstrations were important in the development of the application and the network because we were able to demonstrate, to

an audience, progress of both the application and the network at different stages. Various problems caused by inter-operability between the various components of the MAGIC network were also brought to light at these demonstrations. Without some of these integration tests and demonstrations, many of these issues might have never have been discovered

12/23/2004

4.1 Demonstrations in 1993

The first demonstration of TerraVision occurred at the end of 1993. TerraVision was demonstrated in Kansas City at the Sprint TIOC. This was the first time that all the components of the MAGIC project were brought together on a part of the MAGIC network. The purpose of this first test was to demonstrate the inter-operability of the various components and to test the network.

This first test was instrumental in showcasing the various strengths and weaknesses of the system. Originally, three ISSs were provided by LBNL and placed at Ft. Leavenworth, KU, and at the TIOC. All three could be connected to the Onyx at the TIOC via the MAGIC network. Unfortunately, inherent limitations of the Onyx showed that the highest bandwidth into the end-user application was achieved with one ISS on an FDDI ring and one ISS over the MAGIC network. There were numerous contention problems between the ATM drivers and the SGI graphics system architecture. Augmenting the TIOC Onyx with more processors helped the overall performance of the application, but did not solve the bandwidth limitations.

Even with the problems with the host driver interfaces, it was shown that the architecture of TerraVision was sound. A user could pan and zoom over imagery with the imagery being located both locally via FDDI and remotely over ATM. The building models extracted using SRI's CME system were superimposed in two and three dimensions on top of the terrain.

We also successfully demonstrated a single GATS vehicle being relayed over the MAGIC network and into TerraVision. The vehicle positions were displayed in pseudoreal time on top of the terrain. Along with the building models, a user could see a virtual Ft. Irwin on the screen of the Onyx with the data being brought over a network.

4.2 Demonstrations in 1994

The second demonstration of TerraVision occurred in March 1994 in Sioux Falls, SD. Once again limitations inherent in the Onyx forced us to connect one ISS via FDDI locally and one via ATM over the MAGIC network. The TerraVision application was more mature at this point. We successfully demonstrated a 3D flythrough over the terrain.

This was also the first time that multiple GATS vehicles were demonstrated with the vehicle data arriving over the MAGIC network. The vehicles could be superimposed on the terrain while the user was flying through the terrain.

The most significant outcome of this demonstration was feedback from personnel at the Battle Command Battle Laboratory (BCBL) for enhancements to TerraVision to make it more usable from their viewpoint. These enhancements were incorporated into TerraVision in anticipation of the next demonstration.

A series of demonstrations of TerraVision was performed at the BCBL in April and May of 1994 during the Prairie Warrior exercise. This exercise was an academic vehicle for conducting battle-command-related experiments.

During the two-week period of the Prairie Warrior exercise, SRI personnel provided more than 20 separate briefings and demonstrations of TerraVision on the MAGIC network. These demonstrations were seen by more than 100 high-ranking U.S. military officers and visitors from the United Kingdom, Belgium, and the Republic of Korea.

TerraVision was demonstrated using one ISS connected to the SGI Onyx workstation via ATM on the MAGIC network and another ISS connected locally via FDDI. This was the first demonstration of the large six-scene tile set generated by SRI. GATS was also demonstrated with the vehicle positions being brought

12/23/2004

across the MAGIC network in real time.

In the summer of 1994, TerraVision was demonstrated at the first MAGIC Symposium held in Kansas City. The over-the-shoulder application was demonstrated, as was TeReVision. Although the application was being run at the Sprint TIOC, a user in Lawrence, KS was able to control the application using TeReVision. Simultaneously, another user could view the output using KU's over-the-shoulder application. TerraVision was also demonstrated that summer at the National Armed Forces Communications-Electronics Association (AFCEA) TechNet Conference in Washington DC.

In November 1994, TerraVision was demonstrated at Supercomputing '94 in Washington DC. Connecting back to the MAGIC network via a DS3 link, we were able to showcase TerraVision to outside of the MAGIC network. Once again, TerraVision was locally connected to one ISS via FDDI and one remote ISS located on the MAGIC network.

4.3 Demonstrations in 1995

In 1995, TerraVision was successfully demonstrated at SIGGRAPH '95 in Los Angeles, Supercomputing '95 in San Diego, and the Second MAGIC Symposium in Minneapolis.

At the symposium in Minneapolis, we demonstrated TerraVision on the MAGIC network. Imagry was drawn from several locations across the MAGIC network simultaneously. TerraVision was enhanced to allow two users to simultaneously control and view TerraVision using SRI's XCPS collaboration software. The XCPS system sits on top of the X11 protocol layer and acts as a pseudo server. By intercepting and replicating X events, multiple users can simultaneously view and manipulate the same application.

For SIGGRAPH, TerraVision utilized a DS3 link between Los Angeles and Sioux Falls. At Supercomputing '95, TerraVision utilized the I-Way high-speed link between San Diego and MAGIC. Although there were problems with the network, we were able to successfully demonstrate the capabilities of the system.

At SC '95, TerraVision was enhanced in several ways. Collaborating with LBNL, a 3D tracker and joystick was added. This was accomplished by turning TerraVision into an Application Visualization System (AVS) module, which allowed LBNL to use preexisting software to quickly integrate new input devices. In the future, this will provide a path to quickly add more functionality to TerraVision by adding in new AVS modules as they are developed.

Also at SC '95, TerraVision was enhanced to utilize the Electronic Visualization Laboratory (EVL) CAVE environment. The CAVE environment includes interfaces such as the ImmersaDesk, the CAVE and the NII Wall. TerraVision was successfully demonstrated using the NII Wall and the ImmersaDesk. We were not able to obtain a slot in the CAVE for demonstration purposes.

The NII Wall is a large-scale projection system that seamlessly projects the video output from separate sources to combine them into one very large high-resolution screen. This environment is good for large-scale presentations and briefings. Although TerraVision had a slot for the NII Wall demonstration, a disk crash right before the allocated time cut the actual demonstration to only 10 minutes. Fortunately, we were able to demonstrate TerraVision. In the future, this technology could be used for briefings in a theater-like setting where it is more cost effective for a large-scale audience than for one-on-one visualizations.

The ImmersaDesk utilizes a rear projection system in an architect's workbench setting. The image is back projected onto an angled screen in stereo. The user, wearing stereo glasses and a wand, stands in front of the screen. Both the glasses and the wand have a 3D tracker. This allows the illusion of an immersive environment and interaction by a single user. The ImmersaDesk is mainly used for small groups and single users. We had several slots for the ImmersaDesk and were able to successfully demonstrate it to numerous

12/23/2004

conference attendees. In the future, this type of technology would be ideal for a single visualizing and interacting with a large volume of terrain. It would also be ideal for small-scale briefings.

4.4 Ongoing Demonstrations

TerraVision is currently being used at several locations for demonstrations on an ongoing basis. Sprint uses TerraVision to demonstrate the MAGIC network both internally and externally. NASA Goddard researchers also use TerraVision for demonstration and testing for projects on which they are working. By the close of the project, the application had been used on several high-speed network test-beds, including Spartan, Bagnet, NTONC, ATDNet, and AAI for purposes of traffic measurement and demonstration.

5. Technology Transition

Various activities have been accomplished in transitioning the MAGIC technology to agencies and organizations not directly associated with the Consortium. Such activities included briefings, numerous demonstrations of the technology, and actual porting of portions of the MAGIC software.

- In the early stages of the MAGIC project, technology transition consisted primarily of briefings to other agencies that were believed to have an interest in, or a need for, the technology under investigation in MAGIC. For example, SRI briefed MAGIC to personnel at the Air Force Global Weather Central, Offutt Air Force Base, NE, and to staff at Naval Sea Systems Command (NAVSEA), to solicit their interest in the technology once it was developed.
- Early in the program SRI's Interprocess Communication Protocol software was transferred to LBNL.
- The SRI-developed software that allowed TerraVision to send frames as IP multicast packets was converted to a generic frame-grab utility for SGI machines. This software allows any user attached to the multicast backbone to broadcast frames from any SGI machine so that other users can view them in real time. This software is currently being used by members of the Internet community.
- The Jet Propulsion Laboratory (JPL) expressed interest in incorporating some of the methods TerraVision uses to visualize large databases. JPL later did port TerraVision to its environment. The port allows TerraVision to run independently of the ISS (albeit more slowly), and made TerraVision accessible to a wide range of users in the visualization community.
- SRI demonstrated TerraVision running over the Internet at the JPL, connected to ISSs in Menlo Park. Although this situation was not optimal, it demonstrated the ability of TerraVision and the ISS to adapt to network bandwidth capabilities.
- TerraVision was demonstrated during the 1994 Prairie Warrior Exercise at the BCBL(L), Ft. Leavenworth. In attendance at the numerous demonstrations were 53 visitors, 35 of whom were flag-rank military officers.
- Personnel from the Army TRADOC Analysis Command (TRAC) at Ft. Leavenworth, visited the Sprint Discovery room in Overland Park, KS, where they were given a briefing and a demonstration of TerraVision running on the MAGIC backbone. TRAC has considerable processing and a simulation requirements and is interested in high-speed links to the EROS Data Center and the Army HighPerformance Computing Research Center.
- SRI worked with the BCBL to determine the BCBL's requirements for MAGIC and a plan to use the MAGIC network to support Army initiatives at the BCBL. These requirements center around the use of ACTS T1/T3 satellite terminals and ACTS experiments that focus on the Army's concept of splitbase operations.
- SRI contributed to MAGIC's presentation at the Supercomputing '94 conference held during November 1994. SRI's contribution included planning and installation of the high-speed link from the MAGIC network to show floor, as well as contributions to software, systems, and network installation, testing, and maintenance for the duration of the show. SRI also staffed the booth and presented demonstrations of MAGIC and TerraVision.
- SRI installed the TerraVision software on a system at the Naval Research Laboratory (NRL), and trained an NRL staff member on demonstrating the application.

12/23/2004

- SRI assisted the BCBL in gaining the ability to run TerraVision on one of its SGI Indigos. TerraVision software was installed on an Indigo and the operating system upgraded to IRIX 5.x. The ISS was also installed on the BCBL's SPARCServer 1000.
- SRI worked with the BCBL to utilize the MAGIC backbone to transfer processed LANDSAT and SPOT imagery from the USGS EROS Data Center (EDC) for viewing at the Battle Lab and for redistribution over NASA ACTS low-data-rate terminals to remote exercise locations.
- SRI developed a proposed MAGIC network design, for Ft. Leavenworth, showing where systems will be located as well as the types of interfaces that will be used.
- SRI arranged for Fort Leavenworth's participation in "Army Day" at Sprint's Discovery Room on 22 March 1995. Visitors from Ft. Leavenworth attended demonstrations of applications on the MAGIC and AAI networks. This meeting was an opportunity to consider some of the requirements for MAGIC at Ft. Leavenworth and to develop strategies to retain the network after June 1995.
- SRI created a special turnkey version of TerraVision and installed it in the DARPA Enterprise Room in the fall of 1995.

The following activities occurred at the BCBL, Ft. Leavenworth, during the first half of 1995.

- SRI installed and tested a FORE ATM switch. This switch was used to connect the MAGIC Network to the Battle Lab's SGI Onyx and Sun SPARC 1000 ISS and the Ft. Leavenworth School Network.
- SRI performed a series of demonstrations of TerraVision during the 1995 Prairie Warrior Exercise (PWE). Prairie Warrior serves as a venue for introducing new technological capabilities into the Army, and provides a method for supporting collaborative research efforts.
- SRI worked with an Army Process Action Team (PAT) at Ft. Leavenworth to develop a plan for participation in MAGIC II. The following Ft. Leavenworth activities have expressed interest in participating in MAGIC II in the technology areas indicated.
 - Command and General Staff College (CGSC)--Distance learning
 - Army Knowledge Network (AKN)--Multimedia archives
 - Battle Command Battle Lab (BCBL)--Battlefield visualization
 - TRADOC Analysis Command (TRAC)--Interactive synchronization of warfighting models
 - Communications Electronics Command (CECOM)--Interface with JADE ATM Network
 - Directorate of Information Management (DOIM)--high-speed network access for external customers
 - Ft. Leavenworth School District--Exposure to and interaction with distant school districts
 - National Simulation Center (NSC)--Warfighter's Simulation Development Testbed
 - Medical Department Activity--Telemedicine and teleradiology
- SRI Conducted a series of meetings with Sprint and SBC Communications (formerly Southwestern Bell Telephone) to discuss plans to support some new MAGIC II SONET links into SBC's research facilities in St. Louis, MO, and Austin, TX. These plans have generated a positive interest in MAGIC at SBC Communications. Trademark, Thinking Machines Corp.

12/23/2004